PATENT
Customer No. 22,852
Attorney Docket No. 06502.0194

# IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
## BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: )
)
**Graham HAMILTON** ) Group Art Unit: 2172
)
Application No.: 09/621,687 ) Examiner: J. Fleurantin
)
Filed: July 21, 2000 )
)
For: DATABASE ACCESS BRIDGE ʄ **RECEIVED**
SYSTEM AND PROCESS )

**Mail Stop Appeal Brief--Patents**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

JAN 8 2004

Technology Center 2100

Sir:

## APPEAL BRIEF UNDER 37 C.F.R. § 1.192

In support of its Notice of Appeal filed November 7, 2003, and pursuant to

37 C.F.R. § 1.192, Appellant presents in triplicate this Appeal Brief accompanied by a

check in the amount of $330.00 to satisfy the fee under 37 C.F.R. § 1.17(c). This is an

appeal to the Board of Patent Appeals and Interferences from a decision finally rejecting

claims 1, 2, 4-6, and 8-40. If additional fees are required or if the enclosed payment is

insufficient, please charge the deficiencies to Deposit Account No. 06-0916. If a fee is

required for an extension of time under 37 C.F.R. § 1.136 and such fee is not accounted

for above, Appellant petitions for such an extension and requests that the fee be

charged to Deposit Account No. 06-0916.

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

01/07/2004 JADD01    00000026 09621687
01 FC:1402                330.00 OP

## I.   REAL PARTY IN INTEREST

The real party in interest is the current assignee of the present application, Sun Microsystems, Inc., a corporation of Delaware.

## II.   RELATED APPEALS AND INTERFERENCES

There are no known related pending appeals or interferences directly affected by or having a bearing on the decision in the pending appeal.

## III.   STATUS OF CLAIMS

Claims 1, 2, 4-6, and 8-40 have been finally rejected and are the subject of this appeal. In the Final Office Action dated August 11, 2003, the Examiner rejected claims 1, 2, 4-6, and 8-40 under 35 U.S.C. § 103(a) as being unpatentable over <u>Apte et al.</u> (U.S. Patent No. 6,269,373). The claims on appeal in the form pending after the Final Office Action are set forth in Appendix A at the end of this brief.

## IV.   STATUS OF AMENDMENTS

Appellant files herewith an Amendment under 37 C.F.R. § 1.116, solely to correct minor typographical errors. In particular, the Amendment proposes to correct typographical errors in the specification and in claims 12, 26, 32, and 36. The amended claims are set forth in Appendix B at the end of this brief.

## V.   SUMMARY OF INVENTION

In computer application development, visual builder or application builder tools, which include standard sets of software modules that may be interlinked, are often used to create custom-built applications. Specification at 1, lines 6-9. Such tools allow

2

developers to create database access applications in which a graphical user interface (GUI) facilitates database access through components (e.g. text, buttons, etc.) that are linked to a corresponding database system. Id. at 1, lines 10-21. Many developers have substantial experience with two tier application environments that involve applications designed to directly query or access a database. Id. at 2, lines 1-2. In a two tier environment, a client computer system (first tier) directly communicates with a database system (second tier) to access data. Id. at 2, lines 2-5. With the advent of the Internet and the use of application servers, however, three tier environments are becoming more popular. Id. at 2, lines 6-7.

In a three tier environment, a client system (first tier) has a GUI that communicates with an application running on an application server (second tier) which in turn communicates with a database server (third tier) for access to and storage of data. Id. at 2, lines 7-9. Because three tier environments enable applications on the second tier to be modified without having to substantially modify each client system, a three tier environment may be more efficiently maintained and updated than a two tier environment. Id. at 2, lines 9-13. Developers familiar only with two tier environments, however, are often unable to use many of their existing skills in a three tier environment. Id. at 2, lines 15-19. Instead, developers frequently need to learn new programming protocols or languages, making many businesses reluctant to switch to three tier environments due to training time and cost concerns. Id. at 2, line 19 to 3, line 10. Given the existing skill base for programming in two tier environments, there is a need for a method and system that enables computer program developers to continue to use

3

their existing two tier skills in a three tier environment, without learning new computer programming protocols or languages. Id. at 3, lines 10-13.

Generally described, the systems and processes consistent with the present invention enable a first database call to be converted to a general programming language call, which in turn invokes a second corresponding database call. Id. at 3, lines 16-18. By use of such a system or process, a computer application designed to operate in a two tier computer environment may operate in a three tier computer environment, without specific programming for the three tier computer environment. Id. at 3, lines 18-20.

One aspect consistent with the present invention converts the first database call into a general programming language call using a database bridge map. Id. at 3, line 21 to 4, line 8. This database bridge map is created by analyzing Enterprise Java™ Bean (EJB) components that may be stored at the application server. Id. at 4, lines 9-20. Another aspect consistent with the present invention converts the first database call into a general computer programming language call by exposing software components for accessing a database as elements of the database. Id. at 4, line 21 to 5, line 1. The database call is then converted into a general computer programming language call corresponding to a selected software component. Id. at 5, lines 1-4. A third aspect consistent with the present invention converts the first database call into a general programming language call using a command converter. Id. at 5, lines 5-15.

## VI. ISSUE

The issue in this Appeal is

(1)     Whether the Examiner's rejection of claims 1, 2, 4-6, and 18-40 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. can be affirmed when the Examiner failed to satisfy the initial burden of factually supporting a prima facie case of obviousness in this case because the reference does not teach or suggest each and every recitation of these claims including: (a) receiving an SQL call; (b) mapping an SQL call to a general computer language programming call; (c) receiving a database protocol command; and (d) converting a database protocol command.

## VII. GROUPING OF CLAIMS

In the claims on appeal, claims 1, 19, 26, and 32 are the independent claims. The claims on appeal do not stand or fall together.  These claims should be considered in four groups.

Group I:  1, 2, 4-6, and 8-18;

Group II:  19-25;

Group III:  26-31;

Group IV:  32-40.

The claims have been placed in these groups due to their common subject matter.  Appellant, however, has addressed the outstanding rejections in sections based on the rejections themselves instead of the above identified groupings.

## VIII.  ARGUMENT

### A.  Summary of Arguments

#### 1.  The rejection of claims 1, 2, 4-6, and 8-18.

Appellant appeals the rejection of claims 1, 2, 4-6, and 8-18 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. because the reference fails to teach or suggest each and every recitation of these claims and because the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.  The Examiner's rejection should be reversed for the following reasons.  First, Apte et al. does not teach or suggest the requirement of receiving an SQL call at the computer system, as recited in claim 1.  Apte et al. merely discloses the receipt of object-oriented programming requests.  All use of database protocol language occurs within the server object itself; such commands are neither provided by the client nor received by the server.

Second, Apte et al. does not teach or suggest the requirement of mapping the SQL call to a general computer language programming call of a computer application, as recited in claim 1; there in no infrastructure in Apte et al. for mapping such requests. Third, Apte et al. does not teach or suggest executing the programming call to invoke functions of the computer application that correspond to functions specified by the SQL call.  Fourth, when asked during the interview held on November 6, 2003, the Examiner could not, and would not, point to any particular portion of Apte et al. to support the rejections in the Office Action.  Finally, Apte et al. does not provide any suggestion or motivation that supports the Examiner's conclusion that the recitations of claims 1, 2, 4-6, and 8-18 are obvious.

Claims 2, 4-6 and 8-18 depend upon claim 1, and the rejection of these claims should be reversed for at least the same reasons given with respect to claim 1.

### 2.    The rejection of claims 19-25.

Appellant appeals the rejection of claims 19-25 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. because the reference fails to teach or suggest each and every recitation of these claims and because the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention. The Examiner's rejection should be reversed for the following reasons.

First, Apte et al. fails to teach or suggest the requirement of receiving a database protocol command, as recited in claim 19. Second, Apte et al fails to teach or suggest the requirement of converting said database protocol command into a general computer programming language command for accessing said database, as recited in claim 19. Finally, the Examiner failed to provide any suggestion or motivation to modify Apte et al. in a manner that compensates for the deficiencies of the reference. Claims 20-25 depend upon claim 19, and the rejection of these claims should be reversed for at least the same reasons given with respect to claim 19.

### 3.    The rejection of claims 26-31.

Appellant appeals the rejection of claims 26-31 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. because the reference fails to teach or suggest each and every recitation of these claims and because the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention. The Examiner's rejection should be reversed for the following

reasons. First, Apte et al. fails to teach or suggest the requirement of receiving a database protocol command, as recited in claim 26. Second, Apte et al. fails to teach or suggest the requirement of converting the database protocol command to a command syntax of the first computer programming language corresponding to at least a selected one of the exposed software components, as recited in claim 26. The server object's ability to access other applications does not have anything to do with the use of SQL interfaces, and accordingly does not convert any database protocol commands.

Finally, the Examiner failed to provide any suggestion or motivation to modify Apte et al. in a manner that compensates for the deficiencies of the reference. Claims 27-31 depend upon claim 26, and the rejection of these claims under should be reversed for at least the same reasons given with respect to claim 26.

### 4. The rejection of claims 32-40.

Appellant appeals the rejection of claims 32-40 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. because the reference fails to teach or suggest each and every recitation of these claims and because the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention. The Examiner's rejection should be reversed for the following reasons. First, Apte et al. fails to teach or suggest the requirement of a command converter that converts a database programming language call received from a computer to a general computer programming language call. And, Second, the Examiner failed to provide any suggestion or motivation to modify Apte et al. in a manner that compensates for the deficiencies of the reference. Claims 27-31 depend

upon claim 26, and the rejection of these claims should be reversed for at least the same reasons given with respect to claim 26.

## B.    Summary of Claims

Claim 1 is drawn to a computer-readable medium, operative to serve as a database interface, having instructions which when executed by a computer system, comprise the steps of receiving a Structured Query Logic (SQL) call at the computer system, mapping the SQL call to a general computer language programming call of a computer application, and executing the general computer language programming call to invoke functions of the computer application that correspond to functions specified by the SQL call. Claims 2, 4-6, and 8-18 depend from claim 1 and further define the computer-readable medium recited in claim 1.

Claim 19 is drawn to a computer implemented method for accessing a database that comprises the steps of receiving a database protocol command for accessing elements in a database, converting said database protocol command into a general computer programming language command for accessing said database, and accessing said database by executing said general computer programming language command. Claims 20-25 depend from claim 19 and are drawn to process steps that further define the computer implemented method recited in claim 19.

Claim 26 is drawn to a computer readable medium for accessing a database, comprising instructions which when executed by a computer, comprise the steps of exposing software components, in a first computer programming language, of an application server as database elements, said software components being operative for accessing said database. The claimed instructions, when executed by a computer, also

include the steps of receiving a database protocol command for accessing the database at the application server, converting the database protocol command to a command syntax of the first computer programming language corresponding to at least a selected one of said software components, and accessing said database using said selected one of said software components. Claims 27-31 depend from claim 26 and further define the computer-readable medium recited in claim 26.

Claim 32 is drawn to a system for interfacing between a computer and a database that comprises a command converter operative to convert a first database programming language call received from the computer to a general computer programming language call that corresponds to the database programming language call. Said system is operative to execute said general computer programming language call and operative to generate a second database programming language call, which corresponds to the first database programming language call, to access a database. Claims 33-40 depend from claim 32 and further define the system recited in claim 32.

### C. The Examiner's Rejection of Claims 1, 2, 4-6, and 8-40 Under 35 U.S.C. § 103(a) as Being Unpatentable Over Apte et al. Must be Reversed Because (1) the Reference Does Not Teach or Suggest Each and Every Claim Recitation; and (2) the Examiner Did Not Satisfy the Initial Burden of Factually Supporting a Prima Facie Case of Obviousness.

Appellant respectfully traverses the rejection of claims 1, 2, 4-6, and 8-40 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. because the Examiner has failed to establish a prima facie case of obviousness.

The Examiner's rejection of claims 1, 2, 4-6, and 8-40 should be reversed because Apte et al. does not teach or suggest each and every recitation of these claims, including (1) receiving an SQL call; (2) mapping an SQL call to a general

computer language programming call; (3) receiving a database protocol command; and (4) converting a database protocol command. Furthermore, in addition to the deficiencies of <u>Apte et al.</u> in teaching or suggesting each and every recitation of Appellant's claims, the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.

To establish a prima facie case of obviousness, three basic criteria must be met. First, the prior art reference must teach or suggest each and every element recited in the claims. <u>See</u> M.P.E.P. § 2143.03 (8th ed., rev. Feb. 2003). Second, there must be some suggestion or motivation to modify the reference in a manner resulting in the claimed invention. <u>See</u> M.P.E.P. § 2143. Third, a reasonable expectation of success must exist. <u>See</u> M.P.E.P. § 2143.02. Moreover, each of these requirements must "be found in the prior art, and not based on applicant's disclosure." M.P.E.P. § 2143. In rejecting claims 1, 2, 4-6, and 18-40, the Examiner has failed to meet this criteria.

> **1.      The Examiner's rejection of claims 1, 2, 4-6, and 8-18 under 35 U.S.C. § 103(a) as being unpatentable over <u>Apte et al.</u> must be reversed because (1) <u>Apte et al.</u> does not teach or suggest each and every claim recitation, including receiving an SQL call and mapping the SQL call to a general computer language programming call; and (2) the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.**

Appellant respectfully traverses the rejection of claims 1, 2, 4-6, and 8-18 under 35 U.S.C. § 103(a) as being unpatentable over <u>Apte et al.</u> because the Examiner has failed to establish a prima facie case of obviousness.

The Examiner's rejection of claims 1, 2, 4-6, and 8-18 should be reversed because <u>Apte et al.</u> does not teach or suggest each and every recitation of these claims, including (1) receiving an SQL call at the computer system; (2) mapping the

11

SQL call to a general computer language programming call of a computer application; and (3) executing the programming call to invoke functions of the computer application that correspond to functions specified by the SQL call. Furthermore, in addition to the deficiencies of Apte et al. in teaching or suggesting each and every recitation of these claims, the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.

Appellant wishes to thank the Examiner for the courtesies extended during the interview held on November 6, 2003 with Appellant's representatives, where the Office Action dated August 11, 2003 was discussed. During the interview, Appellant's representatives distinguished the recitations of claims 1, 2, 4-6, and 8-40 from Apte et al. In particular, Appellant's representatives explained the difference between the SQL calls and database protocol commands recited by Appellant, and the object-oriented programming requests taught by Apte et al. The Examiner maintained the claim rejections, despite the arguments presented by Appellant's representatives. The Examiner, however, was unable to identify any disclosure in the reference that teaches or suggests the receipt of SQL calls or database protocol commands. At first, the Examiner rested on the citations provided in the Office action, however, when pressed, the Examiner could not explain why the cited passages taught the claim recitations. Ultimately, when asked by Appellant's representatives, the Examiner could not, and would not, point to any particular portion of Apte et al. that supported his position that the reference taught the receipt of SQL calls or database protocol commands.

Claim 1 recites a combination including, *inter alia*, "receiving a Structured Query Logic (SQL) call at the computer system; mapping the SQL call to a general computer

12

language programming call of a computer application; and executing the programming call to invoke functions of the computer application that correspond to functions specified by the SQL call." The Examiner alleges that <u>Apte et al.</u> teaches receiving an SQL call, but concedes that <u>Apte et al.</u> does not explicitly teach "mapping the SQL call to a general computer language programming call of a computer application; and executing the programming call to invoke functions of the computer application that correspond to functions specified by the SQL call," as recited in claim 1. <u>See</u> Office Action ¶ 3, at 5-6. The Examiner asserts, however, that the reference itself, along with the knowledge generally available to one of ordinary skill in the art, suggest modifying <u>Apte et al.</u> in a manner that compensates for the above noted deficiencies. <u>See</u> <u>Id.</u> Appellant disagrees.

Contrary to Examiner's allegations, <u>Apte et al.</u> does not teach or suggest receiving an SQL call at a server computer system. Instead, the user in <u>Apte et al.</u> submits an object-oriented programming call to a remote server object to invoke a method on that server object. <u>See</u> <u>Apte et al.</u>, col. 7, lines 7-17. A remote method invocation such as this is not the same as an SQL call. The Examiner points specifically to <u>Apte et al.</u>'s disclosure that "[c]lient object 400 may initiate calls to server object 402 to access database 404 based on various business rules or business logic implemented within server object 402," and alleged that this is readable as the receipt of an SQL call at the computer system. <u>Apte et al.</u>, col. 6, lines 31-34; Office Action ¶ 3, at 5. This assertion is mistaken.

Although the identified passage in <u>Apte et al.</u> mentions accessing a database using business rules or business logic, it does not disclose the receipt of an SQL call at

13

the server object. Instead, the passage merely discloses the receipt of an object-oriented programming call to invoke a remote method. Further, the reference discloses that a client may initiate a call to a server object to access a database. See Apte et al., col. 6, lines 31-33. But it is the server object, not the client, that formulates a database access command based on the "business rules or business logic implemented within the server object." Apte et al., col. 6, lines 33-34 (emphasis added). The reference makes no mention of using database rules or logic at the client object and no mention of an SQL call being submitted to the server object. The business rules and/or business logic used to access a database in Apte et al. are contained within the server object and are not provided by the user in the form of an SQL call or any database protocol command. Thus, the server object in Apte et al. does not receive SQL calls from a client, but instead uses database protocol language interfaces to obtain information from the database. Nothing in Apte et al. suggests that the user is submitting anything more than a simple object-oriented programming request to invoke a remote database access method. Accordingly, Apte et al. fails to teach or suggest receiving an SQL call at the computer system, as recited in claim 1.

Further, Apte et al. fails to teach or suggest "mapping the SQL call to a general computer language programming call of a computer application," as recited in claim 1. The Examiner concedes that Apte et al. fails to explicitly disclose this recitation, but asserted that the recitation is implicitly disclosed. See Office Action ¶ 3, at 5-6. The Examiner is incorrect.

There is no infrastructure in Apte et al. to map an SQL call received by the server object. The Examiner asserts that the process disclosed in Apte et al. of mapping

14

references to a CORBA server containing Enterprise Java Beans (EJB's) is equivalent to the claimed mapping step of claim 1. See Apte et al., col. 16, line 66 to col. 17, line 3; Office Action ¶ 3, at 5-6. Appellant disagrees. The Object Request Browsers (ORB's) in Apte et al. invoke a remote method identified by an object-oriented programming call submitted by the user. See Apte et al., col. 7, lines 7-17 & col. 8, lines 10-23. The process disclosed in Apte et al. "flattens" CORBA server references into data strings and stores this information in a back-end data store as a way to persist EJB's. See Apte et al., col. 17, lines 4-39. When an object-oriented programming call is submitted by the client, this back-end data store is used to process the remote method invocation. This process of using object-oriented programming techniques and a back end data store with flattened CORBA references to invoke remote methods is not the same as receiving and mapping an SQL call. Apte et al. fails to show any capabilities for mapping an SQL call received from the client to a general computer language programming call in a manner consistent with the steps recited in claim 1, and therefore fails to teach or suggest this recitation.

Finally, Apte et al. fails to teach or suggest "executing the general computer language programming call to invoke functions of the computer application that correspond to functions specified by the SQL call," as recited in claim 1. The Examiner concedes that Apte et al. fails to explicitly disclose this recitation, but again asserts that the recitation is implicitly disclosed. See Office Action ¶ 3, at 5-6. The Examiner is incorrect. As previously discussed, Apte et al. fails to teach or suggest either the receipt of an SQL call or mapping the SQL call to a general computer language programming call, as recited in claim 1. Consequently, Apte et al. likewise fails to teach or suggest

15

executing the general computer language programming call in a manner specified by

the SQL call, as further recited in claim 1.

Also, Appellant traverses the Examiner's mischaracterization of Appellant's

previous response. In particular, the Examiner incorrectly alleges that "Applicant admits

on page 4, that [the] Apte reference teaches the system in the claimed invention."

Office Action ¶ 2, at 3. This allegation is false. In the response filed June 16, 2003,

Appellant argued:

> Apte et al. teaches a distributed system that allows a client to invoke methods on server objects and receive results from the invocation. In operation, the client sends an object-oriented programming call to a server object, which processes the request by invoking methods identified in the client request. The server object collects response from the invocation and returns them to the client.
>
> In contrast, claim 1 recites receiving an SQL call, mapping the SQL call to a general computer language programming call of a computer application, and executing the programming call to invoke functions of the computer application that correspond to the functions specified by the SQL call. Although Apte et al. states that a client may send calls to a server object to "access a database" (see col. 6, lines 30-35), the client calls are not database structured calls. Instead, the client calls are method invocation requests that are directed to a server object. The Examiner asserts that Apte et al.'s JDBC SQL interfaces used by the server objects teach SQL calls, as recited in claim 1 (see Office Action, page 4, lines 9-12). Applicant disagrees. As mentioned above, the SQL interface is used by a server object to access relational databases. These server objects do not receive SQL calls from a client, but instead use SQL interfaces to obtain information from a database. Therefore, the client does not send, and the server does not receive, SQL calls that are mapped to a general computer language programming call. Instead, the client calls are object-oriented programming calls. For example, Apte et al. describes how a client locates proxy class objects and uses remote method invocation techniques to invoke a business method on an EJB running on a CORBA server (see Fig. 8).

16

> These client based processes do not include providing an
> SQL call to the server.

Amendment and Request for Reconsideration, June 16, 2003, at 4-5.

Nowhere in this response does Appellant admit that <u>Apte et al.</u> teaches the system recited in claim 1. On the contrary, Appellant explicitly traverses the Examiner's allegation that <u>Apte et al.</u> teaches the claimed system. The Examiner's remark is improper and the Board should consider it as such.

Moreover, the Examiner failed to establish a proper suggestion or motivation to modify <u>Apte et al.</u> in a manner resulting in the claimed invention. The Examiner stated that "[i]t would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teachings of Apte with mapping the database call to a general computer language programming call of a computer application; and executing the general computer language programming call to invoke functions of the computer application that correspond to functions specified by the database call." This statement, however, is insufficient to establish a prima facie case of obviousness because it is established that "[t]he level of skill in the art cannot be relied upon to provide the suggestion to combine references." M.P.E.P. § 2143.01, at 2100-125 (citing <u>Al-Site Corp. v. VSI Int'l Inc.</u>, 174 F.3d 1308 (Fed. Cir. 1999)). The Examiner's conclusionary statement is insufficient to show the proper motivation or suggestion required in an obviousness-type rejection. Furthermore, "[a]lthough a prior art device 'may be capable of being modified to run the way the apparatus is claimed, there must be a suggestion or motivation <u>in the reference</u> to do so.'" <u>Id.</u> § 2143.01, at 2100-126 (quoting <u>In re Mills</u>, 916 F.2d 680, 682 (Fed. Cir. 1992)) (emphasis added). As explained above, there is no such suggestion in <u>Apte et al.</u>

Based on the foregoing, Appellant submits that the Examiner has failed to establish a prima facie case of obviousness with respect to claim 1 and accordingly requests that the Examiner's rejection of this claim under 35 U.S.C. § 103(a) be reversed.

Claims 2, 4-6, and 8-18 depend from claim 1. As explained above, claim 1 is patentable over Apte et al. Accordingly, dependent claims 2, 4-6, and 8-18 are also patentable over Apte et al. for at least the reasons given with respect to claim 1. Appellant therefore requests that the Examiner's rejection of claims 2, 4-6, and 8-18 be reversed.

**2.    The Examiner's rejection of claims 19-25 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. must be reversed because (1) Apte et al. does not teach or suggest each and every claim recitation, including receiving a database protocol command and converting said database protocol command; and (2) the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.**

Appellant respectfully traverses the rejection of claims 19-25 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. because the Examiner has failed to establish a prima facie case of obviousness.

The Examiner's rejection of claims 19-25 should be reversed because Apte et al. does not teach or suggest each and every recitation of these claims, including (1) receiving a database protocol command for accessing elements in a database; and (2) converting said database protocol command into a general computer programming language command for accessing said database. Furthermore, in addition to the deficiencies of Apte et al. in teaching or suggesting each and every recitation of these

18

claims, the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.

Claim 19 recites a combination, which includes, *inter alia*, "receiving a database protocol command for accessing elements in a database" and "converting said database protocol command into a general computer programming language command for accessing said database." The Examiner addresses claim 19 in association with claim 1, and in doing so, relies on the analysis of claim 1 in rejecting claim 19. Accordingly, the Examiner appears to be alleging that Apte et al. either explicitly or implicitly teaches all recitations of claim 19, and that the reference itself, along with the knowledge generally available to one of ordinary skill in the art, suggest modifying Apte et al. in a manner that compensates for any deficiencies of the reference. See Office Action ¶ 3, at 5-6. Appellant disagrees.

Contrary to Examiner's allegations, Apte et al. does not teach or suggest at least "receiving a database protocol command for accessing elements in a database," as recited in claim 19. As explained, Apte et al. merely discloses the receipt, by the server object, of object-oriented programming calls to invoke remote methods on the server object. All use of business methods, business logic, and database protocol language occurs within the server object itself; such commands are neither provided by the client nor received by the server. Thus, Apte et al. fails to teach or suggest at least "receiving a database protocol command for accessing elements in a database," as recited in claim 19 (emphasis added).

Moreover, because Apte et al. fails to teach the receipt of a database protocol command, it consequently fails to teach at least "converting said database protocol

19

command into a general computer programming language command for accessing said database," as further recited in claim 19.

Furthermore, the Examiner fails to establish a proper suggestion or motivation to modify Apte et al. in a manner resulting in the claimed invention. As previously explained, a statement that "[i]t would have been obvious to a person of ordinary skill in the art to modify the teachings of Apte," (see Office Action ¶ 3, at 5), is insufficient to establish a prima facie case of obviousness because the level of skill in the art cannot be relied upon to provide the suggestion to modify a reference (see M.P.E.P. § 2143.01, at 2100-125 (citing Al-Site Corp. v. VSI Int'l Inc., 174 F.3d 1308 (Fed. Cir. 1999))). The Examiner's unsupported conclusions that it would have been obvious to modify Apte et al. are insufficient to show the proper motivation required in an obviousness-type rejection.

Based on the foregoing, Appellant submits that the Examiner has failed to establish a prima facie case of obviousness with respect to claim 19 and accordingly requests that the Examiner's rejection of this claim under 35 U.S.C. § 103(a) be reversed.

Claims 20-25 depend from claim 19. As explained above, claim 19 is patentable over Apte et al. Accordingly, dependent claims 20-25 are also patentable over Apte et al. for at least the reasons given with respect to claim 19. Appellant therefore requests that the Examiner's rejection of claims 20-25 be reversed.

### 3. The Examiner's rejection of claims 26-31 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. must be reversed because (1) Apte et al. does not teach or suggest each and every claim recitation, including exposing software components, receiving a database protocol command, and converting said database protocol command; and (2) the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.

Appellant respectfully traverses the rejection of claims 26-31 under

35 U.S.C. § 103(a) as being unpatentable over Apte et al. because the Examiner has

failed to establish a prima facie case of obviousness.

The Examiner's rejection of claims 26-31 should be reversed because Apte et al.

does not teach or suggest each and every recitation of these claims, including

(1) exposing software components, in a first computer programming language, of an

application server as database elements, said software components being operative for

accessing said database; (2) receiving a database protocol command; and

(3) converting the database protocol command to a command syntax of the first

computer programming language corresponding to at least a selected one of said

software components. Furthermore, in addition to the deficiencies of Apte et al. in

teaching or suggesting each and every recitation of these claims, the Examiner failed to

establish a proper suggestion or motivation to modify the reference in a manner

resulting in the claimed invention.

Claim 26 recites a combination, which includes, *inter alia*, "exposing software

components, in a first computer programming language, of an application server as

database elements, said software components being operative for accessing said

database;" "receiving a database protocol command;" and "converting the database

protocol command to a command syntax of the first computer programming language

21

corresponding to at least a selected one of said software components." The Examiner alleges that Apte et al., either explicitly or implicitly teaches all recitations of Appellant's claim 26. Appellant disagrees.

Contrary to Examiner's allegations, Apte et al. does not teach or suggest at least "receiving a database protocol command for accessing the database at the application server," as recited in claim 26. As explained above with respect to claim 19, Apte et al. fails to teach or suggest "receiving a database protocol command," as recited in claim 26 because all use of database protocol language occurs within the server object itself; such commands are neither provided by the client nor received by the server. Thus, Apte et al. fails to teach or suggest at least "receiving a database protocol command for accessing the database at the application server," as recited in claim 26.

Further, Apte et al. fails to teach or suggest at least "exposing software components, in a first computer programming language, of an application server as database elements, said software components being operative for accessing said database," or "converting the database protocol command to a command syntax of the first computer programming language corresponding to at least a selected one of said software components," as recited in claim 26.

The process described with respect to Fig. 4 of Apte et al. (cited by the Examiner) does not teach, suggest, or even mention converting database protocol commands to a command syntax of any language, much less a programming language corresponding to a selected one of the software components exposed as database elements. The Examiner alleges that the ability of server object 402 in Apte et al. to provide access to current enterprise applications (410) and legacy applications (408) as

22

well as the database (404) is readable as accessing said database using said selected one of said software components. See Office Action ¶ 3, at 9. The Examiner's interpretation of Apte et al. is incorrect.

Although the server object in Apte et al. can provide access to other applications in addition to the database, such access has nothing to do with converting database protocol commands to a command syntax. For example, if a client invokes a method relating to either enterprise or legacy applications, then the server object will run the method to access these applications and retrieve the requested data. The server object only uses a database access language, such as a JDBC SQL interface, to access relational databases contained in the database object (404). See Apte et al., col 6, lines 46-51. The server object's ability to access other applications does not have anything to do with that object's use of SQL interfaces, and accordingly does not convert any database protocol commands. Therefore, Apte et al. fails to teach or suggest at least "converting the database protocol command to a command syntax of the first computer programming language corresponding to at least a selected one of said software components," as recited in claim 26.

Moreover, the Examiner failed to establish a proper suggestion or motivation to modify Apte et al. in a manner resulting in the claimed invention. With respect to claim 26, the Examiner identifies no suggestion or motivation, in either the reference itself or the skill in the art, to modify Apte et al. The only statement made by the Examiner regarding a suggestion or motivation to modify Apte et al. was in regard to claims 1 and 19, where the Examiner alleges that "[i]t would have been obvious to a person of ordinary skill in the art to modify the teachings of Apte." See Office Action ¶ 3, at 5. As

23

mentioned above, such a statement is insufficient to establish a prima facie case of obviousness because the level of skill in the art cannot be relied upon to provide the suggestion to modify a reference. See M.P.E.P. § 2143.01, at 2100-125 (citing Al-Site Corp. v. VSI Int'l Inc., 174 F.3d 1308 (Fed. Cir. 1999)).

Based on the foregoing, Appellant submits that the Examiner has failed to establish a prima facie case of obviousness with respect to claim 26 and accordingly requests that the Examiner's rejection of this claim under 35 U.S.C. § 103(a) be reversed.

Claims 27-31 depend from claim 26. As explained above, claim 26 is patentable over Apte et al. Accordingly, dependent claims 27-31 are also patentable over Apte et al. for at least the reasons given with respect to claim 26. Appellant therefore requests that the Examiner's rejection of claims 27-31 be reversed.

### 4. The Examiner's rejection of claims 32-40 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. must be reversed because (1) Apte et al. does not teach or suggest each and every claim recitation, including a command converter operative to convert a first database programming language call received from the computer; and (2) the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.

Appellant respectfully traverses the rejection of claims 32-40 under 35 U.S.C. § 103(a) as being unpatentable over Apte et al. because the Examiner has failed to establish a prima facie case of obviousness.

The Examiner's rejection of claims 32-40 should be reversed because Apte et al. does not teach or suggest each and every recitation of these claims, including a command converter operative to convert a first database programming language call received from the computer to a general computer programming language call that

corresponds to the database programming language call. Furthermore, in addition to the deficiencies of Apte et al. in teaching or suggesting each and every recitation of these claims, the Examiner failed to establish a proper suggestion or motivation to modify the reference in a manner resulting in the claimed invention.

Claim 32 recites a combination, which includes, *inter alia*, "a command converter operative to convert a first database programming language call received from the computer to a general computer programming language call that corresponds to the database programming language call." The Examiner alleges that Apte et al., either explicitly or implicitly, teaches all the recitations of Appellant's claim 32. Appellant disagrees.

Contrary to Examiner's allegations, Apte et al. does not teach or suggest at least "a command converter operative to convert a first database programming language call received from the computer to a general computer programming language call that corresponds to the database programming language call," as recited in claim 32. As previously explained, the calls generated by the client in Apte et al. are not database programming language calls, they are object-oriented programming calls for invoking remote methods from a server object. All use of database protocol language occurs within the server object itself; such commands are neither provided by the client nor received by the server. Additionally, as discussed above, Apte et al. does not teach or suggest a method of converting the database programming language calls. Accordingly, Apte et al. fails to teach or suggest "a command converter operative to convert a first database programming language call received from the computer to a

general computer programming language call that corresponds to the database programming language call," as recited in claim 32.

Moreover, the Examiner failed to establish a proper suggestion or motivation to modify Apte et al. in a manner resulting in the claimed invention. With respect to claim 32, the Examiner identifies no suggestion or motivation, in either the reference itself or the skill in the art, to modify Apte et al. The only statement made by the Examiner regarding a suggestion or motivation to modify Apte et al. was in connection with claims 1 and 19, where the Examiner alleges that "[i]t would have been obvious to a person of ordinary skill in the art to modify the teachings of Apte." See Office Action ¶ 3, at 5. As previously mentioned, such a statement is insufficient to establish a prima facie case of obviousness because the level of skill in the art cannot be relied upon to provide the suggestion to modify a reference. See M.P.E.P. § 2143.01, at 2100-125 (citing Al-Site Corp. v. VSI Int'l Inc., 174 F.3d 1308 (Fed. Cir. 1999)).

Based on the foregoing, Appellant submits that the Examiner has failed to establish a prima facie case of obviousness with respect to claim 32 and accordingly requests that the Examiner's rejection of this claim under 35 U.S.C. § 103(a) be reversed.

Claims 33-40 depend from claim 32. As explained above, claim 32 is patentable over Apte et al. Accordingly, dependent claims 33-40 are also patentable over Apte et al. for at least the reasons given with respect to claim 32. Appellant therefore requests that the Examiner's rejection claims 33-40 be reversed.

## IX.  CONCLUSION

In view of the foregoing, it is respectfully submitted that the final rejection of claims 1, 2, 4-6, and 8-40 should be reversed and such reversal is respectfully requested.

To the extent any extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this Appeal Brief, such extension is hereby respectfully requested.  If there are any fees due under 37 C.F.R. §§ 1.16 or 1.17 which are not enclosed herewith, including any fees required for an extension of time under 37 C.F.R. § 1.136, please charge such fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated:  January 6, 2004

By:_____
Joseph E. Palys
Reg. No. 46,508

Post Office Address (to which
correspondence is to be sent)

Finnegan, Henderson, Farabow,
Garrett & Dunner, L.L.P.
1300 I Street, N.W.
Washington, D.C. 20005
(202) 408-4000

## APPENDIX A

1.      A computer-readable medium, operative to serve as a database interface, having instructions which when executed by a computer system, comprise the following steps:

receiving a Structured Query Logic (SQL) call at the computer system;

mapping the SQL call to a general computer language programming call of a computer application;

executing the general computer language programming call to invoke functions of the computer application that correspond to functions specified by the SQL call.

2.      The medium of claim 1 wherein said general computer language programming call is an Enterprise Java Bean (EJB) call.

3.      (Cancelled).

4.      The medium of claim 1 wherein the computer system is an application server.

5.      The medium of claim 4 wherein the application server receives the SQL call from a client computer system.

6.      The medium of claim 1 further comprising generating a database call to a database in response to executing the general computer language programming call.

7.     (Cancelled).

8.     The medium of claim 6 further comprising validating data operation prior to issuing an SQL call to a database.

9.     The medium of claim 1 wherein the general computer programming language has components that generate database calls to a database in response to executing the general computer language programming calls; and further comprising:

analyzing the components to determine the correspondence between the database elements and the elements of the components that access the database elements; and

creating a database bridge map that identifies the correspondence.

10.     The medium of claim 9 wherein the step of mapping utilizes the database bridge map to map the SQL call to a general programming language call.

11.     The medium of claim 10 wherein the computer programming language is object-oriented and wherein said components are objects.

12.     The medium of claim 11 the step of analyzing the components comprises determining the methods that are invoked in the objects for use in determining said correspondence.

29

13.    The medium of claim 12 wherein the methods are identified by searching for a method of the form <command prefix>XXX.

14.    The medium of claim 13 wherein the <command prefix> is "get".

15.    The medium of claim 13 wherein the <command prefix> is "set".

16.    The medium of claim 12 wherein design patterns are used to map the received SQL call to the general programming call.

17.    The medium of claim 16 wherein said general computer programming language is the Java programming language.

18.    The medium of claim 1, wherein the SQL call received at the computer system is a first SQL database call and a column layout specified in the first SQL database call is different than a second SQL database call generated to a SQL database in response to executing the general computer language programming call.

19.    A computer implemented method for accessing a database, comprising the steps of:

receiving a database protocol command for accessing elements in a database;

converting said database protocol command into a general computer programming language command for accessing said database; and

accessing said database by executing said general computer programming language command.

20.    The method of claim 19 wherein said database protocol command is received from a computer application executing on a first computer system.

21.    The method of claim 20 wherein said computer application is in a different computer programming language than said general computer programming language.

22.    The method of claim 21 wherein said step of converting comprises mapping said database protocol command to a general computer programming language command.

23.    The method of claim 19 wherein said database protocol command is an SQL call.

24.    The method of claim 23 wherein the general computer programming language is the JAVA programming language.

25.    The method of claim 24 wherein said computer application is a Visual Basic computer application.

26. A computer readable medium, for accessing a database, comprising instructions which when executed by a computer, comprises the following steps:

exposing software components, in a first computer programming language, of an application server as database elements, said software components being operative for accessing said database;

receiving database a protocol command for accessing the database at the application server;

converting the database protocol command to a command syntax of the first computer programming language corresponding to at least a selected one of said software components; and

accessing said database using said selected one of said software components.

27. The medium of claim 26 wherein said database protocol command is an SQL command.

28. The medium of claim 26 wherein the first computer programming language is the Java programming language.

29. The medium of claim 26 wherein the first computer programming language has components that generate database calls to a database in response to executing said selected one; and further comprising:

analyzing the components to determine the correspondence between the database elements and the elements of the components that access the database elements;

creating a database bridge map that identifies the correspondence.

30.     The medium of claim 29 wherein the computer programming language is object-oriented and wherein said components are objects.

31.     The medium of claim 30 wherein the step of analyzing the components comprises determining the methods that are invoked in the objects for use in determining said correspondence.

32.     A system for interfacing between a computer and a database, comprising:

a command converter operative to convert a first database programming language call received from the computer to a general computer programming language call that corresponds to the database programming language call; and

said system operative to execute said general computer programming language call and operative to generate a second database programming language call, which corresponds to the first database programming language call, to access a database.

33.     The system of claim 32 wherein said system is an application server.

34.     The system of claim 33 wherein said computer is a client computer that generates said first database programming language call.

35.     The system of claim 34 wherein said database is accessed through a database server.

36.     The system of claims 32 wherein said first and second database programming language calls are SQL calls.

37.     The system of claim 36 wherein said first and second database programming language calls specify different column names.

38.     The system of claim 36 wherein said general programming language is an object-oriented computer programming language.

39.     The system of claim 37 wherein said programming language is the Java programming language.

40.     The system of claim 32 wherein said command converter comprises a mapping module that maps said first database programming language call to said general computer programming language call.

# APPENDIX B

1.      A computer-readable medium, operative to serve as a database interface, having instructions which when executed by a computer system, comprise the following steps:

receiving a Structured Query Logic (SQL) call at the computer system;

mapping the SQL call to a general computer language programming call of a computer application;

executing the general computer language programming call to invoke functions of the computer application that correspond to functions specified by the SQL call.


2.      The medium of claim 1 wherein said general computer language programming call is an Enterprise Java Bean (EJB) call.


3.      (Cancelled).


4.      The medium of claim 1 wherein the computer system is an application server.


5.      The medium of claim 4 wherein the application server receives the SQL call from a client computer system.

6.      The medium of claim 1 further comprising generating a database call to a database in response to executing the general computer language programming call.

7.    (Cancelled).

8.    The medium of claim 6 further comprising validating data operation prior to issuing an SQL call to a database.

9.    The medium of claim 1 wherein the general computer programming language has components that generate database calls to a database in response to executing the general computer language programming calls; and further comprising:

analyzing the components to determine the correspondence between the database elements and the elements of the components that access the database elements; and

creating a database bridge map that identifies the correspondence.

10.    The medium of claim 9 wherein the step of mapping utilizes the database bridge map to map the SQL call to a general programming language call.

11.    The medium of claim 10 wherein the computer programming language is object-oriented and wherein said components are objects.

12.    The medium of claim 11 wherein the step of analyzing the components comprises determining the methods that are invoked in the objects for use in determining said correspondence.

13. The medium of claim 12 wherein the methods are identified by searching for a method of the form <command prefix>XXX.

14. The medium of claim 13 wherein the <command prefix> is "get".

15. The medium of claim 13 wherein the <command prefix> is "set".

16. The medium of claim 12 wherein design patterns are used to map the received SQL call to the general programming call.

17. The medium of claim 16 wherein said general computer programming language is the Java programming language.

18. The medium of claim 1, wherein the SQL call received at the computer system is a first SQL database call and a column layout specified in the first SQL database call is different than a second SQL database call generated to a SQL database in response to executing the general computer language programming call.

19. A computer implemented method for accessing a database, comprising the steps of:

receiving a database protocol command for accessing elements in a database;

converting said database protocol command into a general computer programming language command for accessing said database; and

accessing said database by executing said general computer programming language command.

20. The method of claim 19 wherein said database protocol command is received from a computer application executing on a first computer system.

21. The method of claim 20 wherein said computer application is in a different computer programming language than said general computer programming language.

22. The method of claim 21 wherein said step of converting comprises mapping said database protocol command to a general computer programming language command.

23. The method of claim 19 wherein said database protocol command is an SQL call.

24. The method of claim 23 wherein the general computer programming language is the JAVA programming language.

25. The method of claim 24 wherein said computer application is a Visual Basic computer application.

26.    A computer readable medium, for accessing a database, comprising instructions which when executed by a computer, comprise the following steps:

exposing software components, in a first computer programming language, of an application server as database elements, said software components being operative for accessing said database;

receiving a database protocol command for accessing the database at the application server;

converting the database protocol command to a command syntax of the first computer programming language corresponding to at least a selected one of said software components; and

accessing said database using said selected one of said software components.

27.    The medium of claim 26 wherein said database protocol command is an SQL command.

28.    The medium of claim 26 wherein the first computer programming language is the Java programming language.

29.    The medium of claim 26 wherein the first computer programming language has components that generate database calls to a database in response to executing said selected one; and further comprising:

analyzing the components to determine the correspondence between the database elements and the elements of the components that access the database elements;

creating a database bridge map that identifies the correspondence.

30. The medium of claim 29 wherein the computer programming language is object-oriented and wherein said components are objects.

31. The medium of claim 30 wherein the step of analyzing the components comprises determining the methods that are invoked in the objects for use in determining said correspondence.

32. A system for interfacing between a computer and a database, comprising:

a command converter operative to convert a first database programming language call received from the computer to a general computer programming language call that corresponds to the database programming language call; and wherein

said system is operative to execute said general computer programming language call and operative to generate a second database programming language call, which corresponds to the first database programming language call, to access a database.

33. The system of claim 32 wherein said system is an application server.

34. The system of claim 33 wherein said computer is a client computer that generates said first database programming language call.

35. The system of claim 34 wherein said database is accessed through a database server.

36. The system of claim 32 wherein said first and second database programming language calls are SQL calls.

37. The system of claim 36 wherein said first and second database programming language calls specify different column names.

38. The system of claim 36 wherein said general programming language is an object-oriented computer programming language.

39. The system of claim 37 wherein said programming language is the Java programming language.

40. The system of claim 32 wherein said command converter comprises a mapping module that maps said first database programming language call to said general computer programming language call.

$\cancel{\#10}$

$\cancel{AF}$M

$\cancel{2172}$

# BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

| | |
|---|---|
| In re Application of: | ) |
| | ) |
| **Graham HAMILTON** | ) Group Art Unit: 2172 |
| | ) |
| Application No.: 09/621,687 | ) Examiner: J. Fleurantin |
| | ) |
| Filed: July 21, 2000 | ) |
| | ) |
| For: DATABASE ACCESS BRIDGE | ) |
| SYSTEM AND PROCESS | ) |

**RECEIVED**

JAN 8 2004

Technology Center 2100

**Mail Stop Appeal Brief--Patents**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

## TRANSMITTAL OF APPEAL BRIEF PURSUANT TO 37 C.F.R. § 1.192

Transmitted herewith in triplicate is the APPEAL BRIEF in this application with

respect to the Notice of Appeal filed on November 7, 2003.

This application is on behalf of

☐ Small Entity  ☒ Large Entity

Pursuant to 37 C.F.R. 1.17(f), the fee for filing the Appeal Brief is:

☐ $165.00 (Small Entity)

☒ $330.00 (Large Entity)

☒ Enclosed is a check for $330.00 to cover the above fees.

<u>PETITION FOR EXTENSION.</u> If any extension of time is necessary for the filing of this

Appeal Brief, and such extension has not otherwise been requested, such an extension

01/07/2004 JADDD1  00000026 09621687
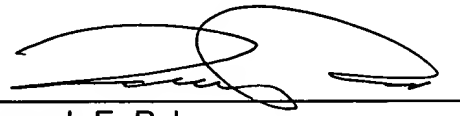
01 FC:1402  330.00 OP

is hereby requested, and the Commissioner is authorized to charge necessary fees for such an extension to our Deposit Account No. 06-0916. A duplicate copy of this paper is enclosed for use in charging the deposit account.

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: January 6, 2004          By: _____
                                    Joseph E. Palys
                                    Reg. No. 46,508